

REMARKS

Applicant is in receipt of the Office Action mailed May 20, 2004. Claim 1, 23, 25, 26, and 29 have been amended. Claims 9-11 have been cancelled. New claims 32-40 have been added. Claims 1-8, and 12-40 remain pending in the case. Reconsideration of the present case is earnestly requested in light of the following remarks.

Section 102 Rejections

Claims 1-10, 12-17, and 19-31 were rejected under 35 U.S.C. 102(e) as being anticipated by Uczekaj et al. (US 5,920,718, "Uczekaj"). Applicant respectfully disagrees. Note that claims 9-11 have been cancelled, and so the rejection of claims 9 and 10 are rendered moot.

As the Examiner is certainly aware, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Amended claim 1 recites:

1. (Currently Amended) A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising:

receiving state diagram information, wherein the state diagram information represents the state diagram and specifies a plurality of states;

programmatically generating the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the plurality of states, wherein the graphical source code comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable by a computer.

The Office Action indicated that Applicant's argument that the reference fails to teach "a graphical program, where a graphical program includes a plurality of interconnected nodes that visually indicate the functionality of the program" did not overcome the rejection because the features relied upon were not recited in the rejected claims. Applicant has amended the independent claims to explicitly include this feature, and to further specify that the generated "graphical program is executable by a computer", and so respectfully submits that Uczekaj does not teach or suggest the invention as represented by the amended independent claims and those claims respectively dependent thereon.

Furthermore, the Office Action asserts that Uczekaj teaches "generating a graphical program based on a state diagram", citing col. 10, lines 16-40, col. 3, lines 33-49, and col 4, lines 28-39. Applicant respectfully disagrees.

Applicant submits that Uczekaj fails to teach numerous features of amended claim 1. For example, as noted in the prior response, Uczekaj fails to teach generating a graphical program at all "wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and wherein the graphical program is executable by a computer" (as defined in claim 1, and in the specification on page 3, lines 6-10, and pages 7-13), and more specifically fails to teach or suggest generating a graphical program based on state diagram information, e.g., a state diagram. Figures 8, 10, 12, 14, 16, 17 and 18 of the present specification provide examples of a graphical program. The Uczekaj reference simply does not teach or suggest a graphical program. Rather, Uczekaj teaches generation of program shell code (in a text-based programming language) for different operating systems. As Uczekaj states in the cited passage,

"The generated code is called application shell code because all code is generated except the specific code for any user method names entered in define user method section 274 and any transition method names entered in define transition method section 284. The specific code for these two types of methods must be entered into specific locations within the application shell code. The user may either enter user and transition method code in the graphical interface tool through an edit function or

outside the graphical interface tool through a file editor that allows a programmer to edit and enter code.” (col. 10, lines 27-37)

Thus, in the system of Uczekaj, the actual program functionality must be manually written by the user, i.e., only the *shell* of the program is generated programmatically, and the program shell is itself written in a text-based programming language. Uczekaj is quite clear in making the distinction between an application (program) and an application shell (program shell), as may be seen in the paragraph cited above. Thus, not only is Uczekaj’s program shell code not a graphical program, but it is also not executable to perform the functionality, but rather requires the user to provide this functionality.

Applicant further notes Uczekaj’s “graphical control system” and “graphical interface tool” refer to a tool with a graphical user interface, where the tool receives user input to the interface, and generates an application shell, where the application shell is in a text-based language. Applicant respectfully submits that the Examiner’s assertion that generating text-based shell code “by activating a generate code button within the graphical interface tool window” somehow makes the generated text-based application shell a “graphical program” is improper, and thus submits that the Examiner has incorrectly described the generated shell as a graphical program.

Regarding the Examiner’s assertion that Uczekaj teaches programmatically generating a graphical program “based on a state diagram”, Applicant notes that Uczekaj actually discloses generating and displaying a state diagram for a class based on user input specifying the class, and that the program shell is generated based on this same user input, *not* the state diagram (col. 10, lines 7-40). In other words, the state diagram generated is that of a class, not a program. Moreover, as noted above, no graphical program is generated at all in Uczekaj, but only a text-based program shell, to which the user must add his own (text-based) program code.

As noted above, Applicant submits that in Uczekaj, the user input used to generate the program shell and state diagram does *not* comprise the actual functionality of user methods and transition methods, and that this functionality, i.e., program code, must be provided by the user. While this user input is used to generate the application shell code, it is not, in fact, used to programmatically generate a graphical program.

Moreover, unlike the graphical program of the present claims, which is executable to perform the functionality of the program, Uczekaj's generated program shell code is specifically *not* executable to perform the functionality, since this functionality must be provided by the user.

Thus, Uczekaj does not teach or suggest programmatically generating a graphical program in response to state diagram information, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate operation of the graphical program, and wherein the graphical program is executable by a computer, and so Applicant submits that claim 1, and claims dependent thereon, are patentably distinct over Uczekaj and unobvious, and are thus allowable for at least the reasons provided above.

Independent claims 23, 25, 26, and 29 include similar limitations as claim 1, and so the arguments presented above apply with equal force to these claims. Thus, for at least the reasons provided above, Applicant submits that claims 23, 25, 26, and 29, and claims respectively dependent thereon, are patentably distinct and unobvious over Uczekaj, and are thus allowable.

Removal of the 102 rejection of claims 1-8, 12-17, and 19-31 is respectfully requested. Applicant further submits that the Uczekaj does not suggest or render obvious any of the pending claims for at least the reasons given above.

Section 103 Rejections

Claims 11 and 18 were rejected under 35 U.S.C. 103(a) as being unpatentable over Uczekaj et al. (US 5,920,718, "Uczekaj"), further in view of Kodosky et al. (US 5,732,277, "Kodosky"). Applicant respectfully disagrees. Applicant also notes that since these claims depend from allowable independent claims (as currently amended), as argued above, claims 11 and 18 are similarly allowable as currently presented. Additional arguments are provided in the previous Office Action Response of March 4, 2004, which is hereby incorporated by reference.

Thus, for at least the reasons provided, Applicant respectfully submits that neither Uczekaj nor Kodosky, either singly or in combination, teaches all of the limitations of claim 11 and claim 18, and so Applicant submits that claims 1 and 18 are non-obvious over Uczekaj in view of Kodosky, and is thus allowable. Removal of the 103 rejection of claims 11 and 18 is respectfully requested.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

New Claims

As noted above, new claims 32-35 have been added. Applicant notes that some of the arguments presented above also apply to new independent claim 32, which recites:

32. (New) A computer-implemented method for programmatically generating a graphical program based on state diagram information, comprising:

receiving the state diagram information, wherein the state diagram information specifies a plurality of states and transitions between the states;

programmatically generating the graphical program in response to the state diagram information, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein a first one or more nodes comprise graphical source code executable to implement first functionality corresponding to a first one or more states, and where a second one or more nodes are user-configurable to implement second functionality of a corresponding second one or more states.

More specifically, Applicant submits that Uczekaj does not teach or suggest programmatically generating a graphical program, or even a graphical program shell, noting that Uczekaj's application shell code is written in a text-based language, and is specifically *not* a graphical program as defined in claim 32 and the specification.

Additionally, neither Uczekaj nor Kodosky teaches or suggests programmatically generating a graphical program in response to state diagram information as recited above.

Thus, Applicant submits that neither Uczekaj nor Kodosky, either singly or in combination teaches the limitations of claim 32, and so claim 32 and those claims dependent thereon are patentably distinct over the cited art and are thus allowable.

Obviousness-Type Double Patenting Rejection

Claims 1-10, 12-15, and 20-31 were provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims of application No. 09/745023. Applicant respectfully traverses this double-patenting rejection at least with respect to some of the claims. However applicant respectfully requests that this double-patenting rejection be held in abeyance until the claims are otherwise indicated as allowable, at which time Applicant will consider the filing of a terminal disclaimer to obviate the rejection. Applicant notes that the filing of terminal disclaimer to obviate a rejection based on non-statutory type double patenting is not an admission of the propriety of the rejection. (See, e.g., MPEP 804.02).

CONCLUSION

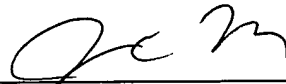
Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-45900/JCH.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☒ Information Disclosure Statement
- ☒ Request For Continued Examination

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: August 20, 2004 JCH/MSW